Renesas Electronics America Inc - M30281FATHP#U3AAE8 Datasheet

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	-
Core Size	-
Speed	-
Connectivity	-
Peripherals	-
Number of I/O	-
Program Memory Size	-
Program Memory Type	-
EEPROM Size	-
RAM Size	-
Voltage - Supply (Vcc/Vdd)	-
Data Converters	-
Oscillator Type	-
Operating Temperature	-
Mounting Type	-
Package / Case	-
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/m30281fathp-u3aae8

Figure 1.2 is a block diagram of the M16C/28 Group (T-ver./V-ver.), 64-pin package.

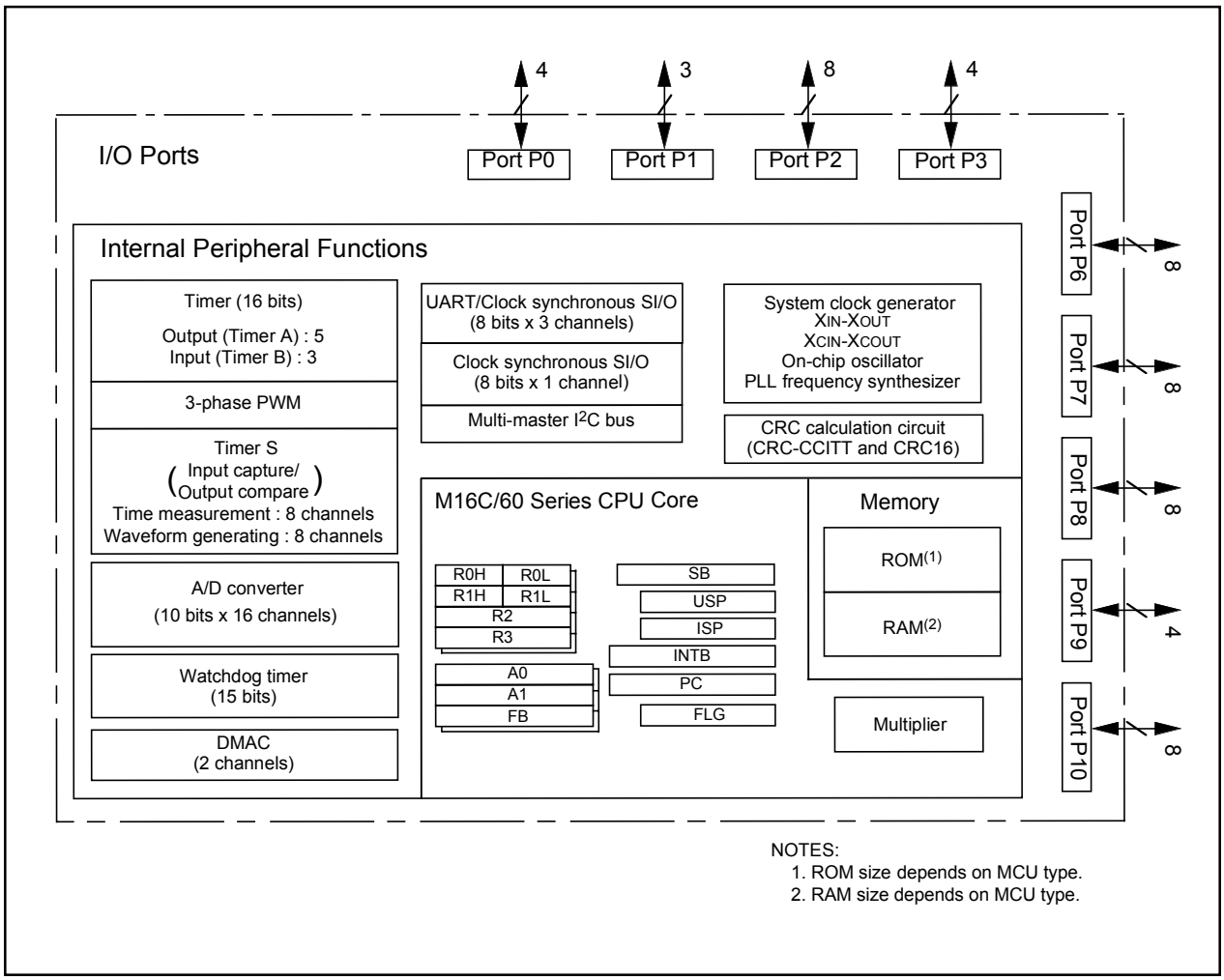


Figure 1.2 M16C/28 Group (T-ver./V-ver.), 64-Pin Package Block Diagram

7. Clock Generation Circuits

The MCU has four clock generation circuits as follows:

(1) Main clock oscillation circuit

(2) Sub clock oscillation circuit

(3) On-chip oscillator

(4) PLL frequency synthesizer

Table 7.1 lists the specifications of the clock generation circuit. **Figure 7.1** shows the clock generation circuits. **Figures 7.2** to **7.7** show the clock-associated registers.

Table 7.1 Clock Generation Circuit Specifications

Item	Main Clock Oscillation Circuit	Sub Clock Oscillation Circuit	Variable On-chip Oscillator	PLL Frequency Synthesizer
Use of clock	- CPU clock source - Peripheral function clock source	- CPU clock source - Timer A, B's clock source	- CPU clock source - Peripheral function clock source - CPU and peripheral function clock sources when the main clock stops oscillating	- CPU clock source - Peripheral function clock source
Clock frequency	0 to 20 MHz	32.768 kHz	Selectable source frequency: $f_{1(ROC)}$, $f_{2(ROC)}$, $f_{3(ROC)}$ Selectable divider: by 2, by 4, by 8	10 to 20 MHz
Usable oscillator	- Ceramic oscillator - Crystal oscillator	- Crystal oscillator	———	———
Pins to connect oscillator	XIN, XOUT	XCIN, XCOUT	———	———
Oscillation stop, restart function	Available	Available	Available	Available
Oscillator status after reset	Oscillating	Stopped	Oscillating (CPU clock source)	Stopped
Other	Externally derived clock can be input		———	———

7.8.1 Operation When CM27 bit = 0 (Oscillation Stop Detection Reset)

When main clock stop is detected when the CM20 bit is 1 (oscillation stop, re-oscillation detection function enabled), the MCU is initialized, coming to a halt (oscillation stop reset; refer to “SFR”, “Reset”). This status is reset with hardware reset 1. Also, even when re-oscillation is detected, the MCU can be initialized and stopped; it is, however, necessary to avoid such usage. (During main clock stop, do not set the CM20 bit to 1 and the CM27 bit to 0.)

7.8.2 Operation When CM27 bit = 1 (Oscillation Stop and Re-oscillation Detect Interrupt)

When the main clock corresponds to the CPU clock source and the CM20 bit is 1 (oscillation stop and re-oscillation detect function enabled), the system is placed in the following state if the main clock comes to a halt:

- Oscillation stop and re-oscillation detect interrupt request occurs.
- The on-chip oscillator starts oscillation, and the on-chip oscillator clock becomes the CPU clock and clock source for peripheral functions in place of the main clock.
- CM21 bit = 1 (on-chip oscillator clock for CPU clock source)
- CM22 bit = 1 (main clock stop detected)
- CM23 bit = 1 (main clock stopped)

When the PLL clock corresponds to the CPU clock source and the CM20 bit is 1, the system is placed in the following state if the main clock comes to a halt: Since the CM21 bit remains unchanged, set it to 1 (on-chip oscillator clock) inside the interrupt routine.

- Oscillation stop and re-oscillation detect interrupt request occurs.
- CM22 bit = 1 (main clock stop detected)
- CM23 bit = 1 (main clock stopped)
- CM21 bit remains unchanged

When the CM20 bit is 1, the system is placed in the following state if the main clock re-oscillates from the stop condition:

- Oscillation stop and re-oscillation detect interrupt request occurs.
- CM22 bit = 1 (main clock re-oscillation detected)
- CM23 bit = 0 (main clock oscillation)
- CM21 bit remains unchanged

7.8.3 How to Use Oscillation Stop and Re-oscillation Detect Function

- The oscillation stop and re-oscillation detect interrupt shares the vector with the watchdog timer interrupt. If the oscillation stop, re-oscillation detection and watchdog timer interrupts both are used, read the CM22 bit in an interrupt routine to determine which interrupt source is requesting the interrupt.
- Where the main clock re-oscillated after oscillation stop, return the main clock to the CPU clock and peripheral function clock source by program. **Figure 7.13** shows the procedure for switching the clock source from the on-chip oscillator to the main clock.
- Simultaneously with oscillation stop, re-oscillation detection interrupt occurrence, the CM22 bit becomes 1. When the CM22 bit is set at 1, oscillation stop, re-oscillation detection interrupt are disabled. By setting the CM22 bit to 0 by program, oscillation stop, re-oscillation detection interrupt are enabled.
- If the main clock stops during low speed mode where the CM20 bit is 1, an oscillation stop, re-oscillation detection interrupt request is generated. At the same time, the on-chip oscillator starts oscillating. In this case, although the CPU clock is derived from the sub clock as it was before the interrupt occurred, the peripheral function clocks now are derived from the on-chip oscillator clock.
- To enter wait mode while using the oscillation stop, re-oscillation detection function, set the CM02 bit to 0 (peripheral function clocks not turned off during wait mode).
- Since the oscillation stop, re-oscillation detection function is provided in preparation for main clock stop due to external factors, set the CM20 bit to 0 (Oscillation stop, re-oscillation detection function disabled) where the main clock is stopped or oscillated by program, that is where the stop mode is selected or the CM05 bit is altered.
- This function cannot be used if the main clock frequency is 2 MHz or less. In that case, set the CM20 bit to 0.

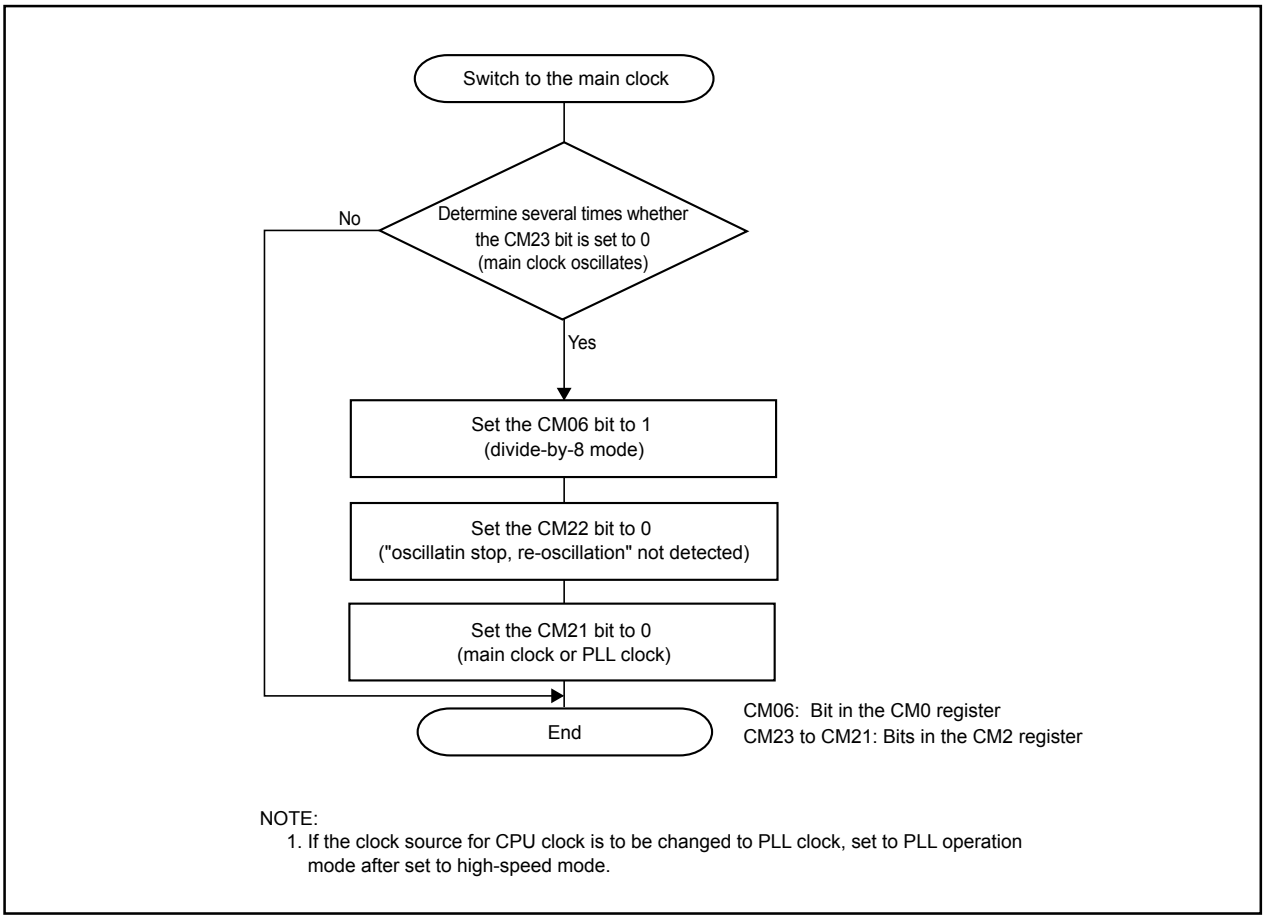


Figure 7.13 Procedure to Switch Clock Source From On-chip Oscillator to Main Clock

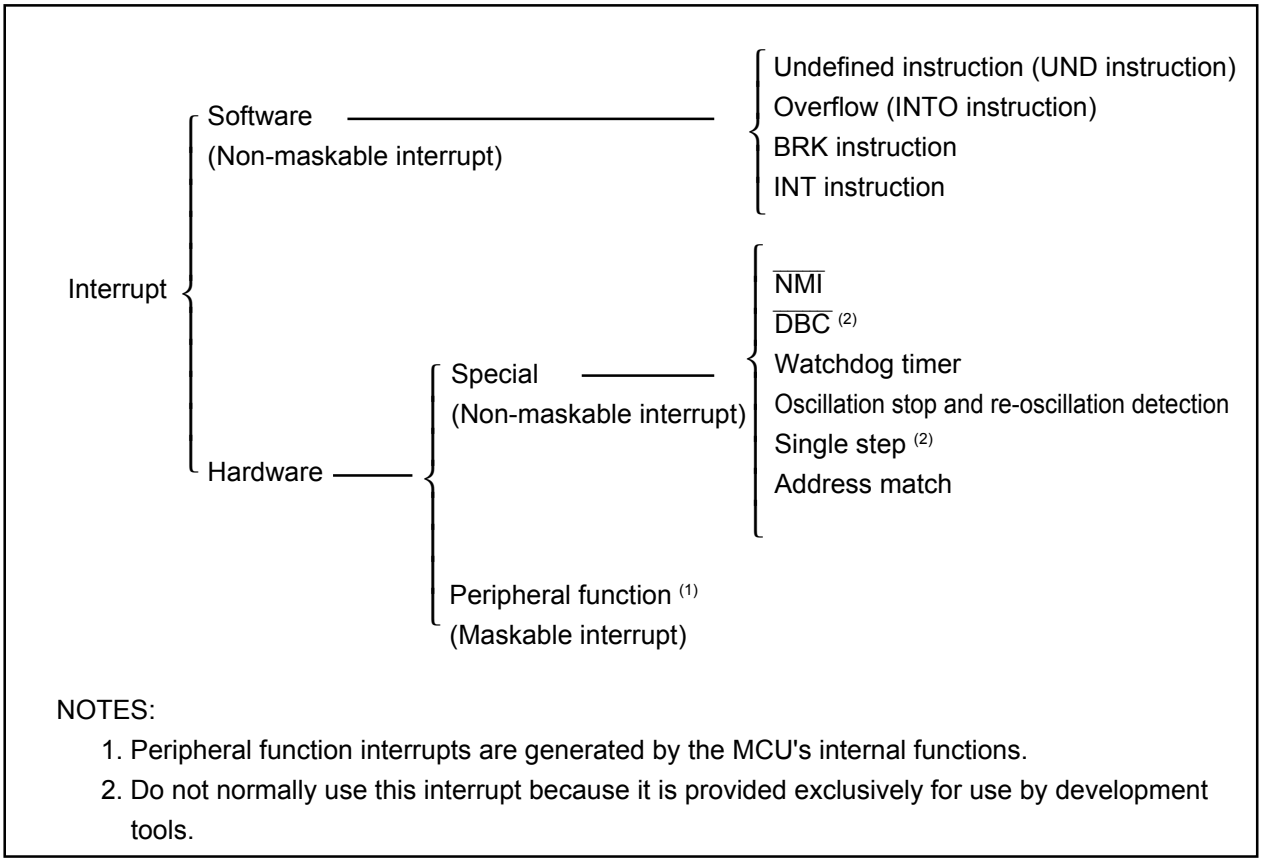
9. Interrupts

Note

The SI/O4 interrupt of peripheral function interrupts is not available in the 64-pin package.

9.1 Type of Interrupts

Figure 9.1 shows types of interrupts.

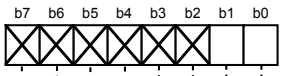


NOTES:
1. Peripheral function interrupts are generated by the MCU's internal functions.
2. Do not normally use this interrupt because it is provided exclusively for use by development tools.

Figure 9.1 Interrupts

- Maskable Interrupt: An interrupt which can be enabled (disabled) by the interrupt enable flag (I flag) or whose interrupt priority **can be changed** by priority level.
- Non-maskable Interrupt: An interrupt which cannot be enabled (disabled) by the interrupt enable flag (I flag) or whose interrupt priority **cannot be changed** by priority level.

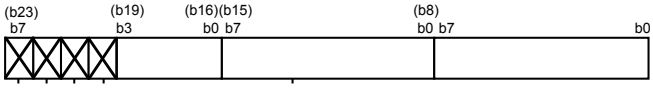
Address Match Interrupt Enable Register



Symbol	Address	After Reset
AIER	0009_{16}	$XXXXXX00_2$

Bit Symbol	Bit Name	Function	RW
AIER0	Address match interrupt 0 enable bit	0: Interrupt disabled 1: Interrupt enabled	RW
AIER1	Address match interrupt 1 enable bit	0: Interrupt disabled 1: Interrupt enabled	RW
$\overline{\text{(b7-b2)}}$	Nothing is assigned. If necessary, set to 0. When read, the content is undefined		—

Address Match Interrupt Register i (i = 0 to 1)



Symbol	Address	After Reset
RMAD0	0012_{16} to 0010_{16}	$X0000_{16}$
RMAD1	$0016_{16}$to 0014_{16}	$X0000_{16}$

Function	Setting Range	RW
Address setting register for address match interrupt	00000_{16} to $FFFFF_{16}$	RW
Nothing is assigned. If necessary, set to 0. When read, the content is undefined		—

Figure 9.13 AIER Register, RMAD0 and RMAD1 Registers

11.1 Transfer Cycles

The transfer cycle consists of a memory or SFR read (source read) bus cycle and a write (destination write) bus cycle. The number of read and write bus cycles is affected by the source and destination addresses of transfer. Furthermore, the bus cycle itself is extended by a software wait.

11.1.1 Effect of Source and Destination Addresses

If the transfer unit is 16 bits and the source address of transfer begins with an odd address, the source read cycle consists of one more bus cycle than when the source address of transfer begins with an even address.

Similarly, if the transfer unit is 16 bits and the destination address of transfer begins with an odd address, the destination write cycle consists of one more bus cycle than when the destination address of transfer begins with an even address.

11.1.2 Effect of Software Wait

For memory or SFR accesses in which one or more software wait states are inserted, the number of bus cycles required for that access increases by an amount equal to software wait states.

Figure 11.5 shows the example of the cycles for a source read. For convenience, the destination write cycle is shown as one cycle and the source read cycles for the different conditions are shown. In reality, the destination write cycle is subject to the same conditions as the source read cycle, with the transfer cycle changing accordingly. When calculating transfer cycles, take into consideration each condition for the source read and the destination write cycle, respectively. For example, when data is transferred in 16 bit units and when both the source address and destination address are an odd address ((2) in **Figure 11.5**), two source read bus cycles and two destination write bus cycles are required.

Three-phase PWM Control Register 0 (1)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	After Reset
INVC0	034816	0016

Bit Symbol	Bit Name	Function	RW
INV00	Effective interrupt output polarity select bit(3)	0: ICTB2 counter is incremented by 1 on the rising edge of timer A1 reload control signal 1: ICTB2 counter is incremented by 1 on the falling edge of timer A1 reload control signal	RW
INV01	Effective interrupt output specification bit(2, 3)	0: ICTB2 counter incremented by 1 at a timer B2 underflow 1: Selected by INV00 bit	RW
INV02	Mode select bit(4)	0: Three-phase motor control timer function unused 1: Three-phase motor control timer function (5)	RW
INV03	Output control bit(6)	0: Three-phase motor control timer output disabled (5) 1: Three-phase motor control timer output enabled	RW
INV04	Positive and negative phases concurrent output disable bit	0: Simultaneous active output enabled 1: Simultaneous active output disabled	RW
INV05	Positive and negative phases concurrent output detect flag	0: Not detected yet 1: Already detected (7)	RW
INV06	Modulation mode select bit(8)	0: Triangular wave modulation mode 1: Sawtooth wave modulation mode (9)	RW
INV07	Software trigger select bit	Setting this bit to 1 generates a transfer trigger. If the INV06 bit is 1, a trigger for the dead time timer is also generated. The value of this bit when read is 0	RW

NOTES:
1. Write to this register after setting the PRC1 bit in the PRCR register to 1 (write enable). Note also that bits INV00 to INV02, bits INV04 and INV06 can only be rewritten when timers A1, A2, A4 and B2 are idle.
2. If this bit needs to be set to 1, set any value in the ICTB2 register before writing to it.
3. Effective when the INV11 bit in the INV1 register is 1 (three-phase mode 1). If INV11 is set to 0 (three-phase mode 0), the ICTB2 counter is incremented by 1 each time the timer B2 underflows, regardless of whether the INV00 and INV01 bits are set. When setting the INV01 bit to 1, the first interrupt is generated when the timer B2 underflows n-1 times, if n is the value set in the ICTB2 counter. Subsequent interrupts are generated every n times the timer B2 underflow.
4. Setting the INV02 bit to 1 activates the dead time timer, U/V/W-phase output control circuits and ICTB2 counter.
5. When the INV02 bit is set to 1 and the INV03 bit is set to 0, $\overline{U}$, U, $\overline{V}$, V, $\overline{W}$, W pins, including pins shared with other output functions, enter a high-impedance state. When INV03 is set to 1, U/V/W corresponding pins generate the three-phase PWM output.
6. The INV03 bit is set to 0 in the following cases:
   - When reset
   - When positive and negative go active (INV05 = 1) simultaneously while INV04 bit is 1
   - When set to 0 by program
   - When input on the $\overline{SD}$ pin changes state from “H” to “L” regardless of the value of the INVCR1 bit. (The INV03 bit cannot be set to 1 when $\overline{SD}$ input is “L”.) INV03 is set to 0 when both bits INV05 and INV04 are set to 1.

Item	INV06=0	INV06=1
Mode	Triangular wave modulation mode	Sawtooth wave modulation mode
Timing at which transferred from registers IDB0 to IDB1 to three-phase output shift register	Transferred only once synchronously with the transfer trigger after writing to registers IDB0 to IDB1	Transferred every transfer trigger
Timing at which dead time timer trigger is generated when INV16 bit is 0	Synchronous with the falling edge of timer A1, A2, or A4 one-shot pulse	Synchronous with the transfer trigger and the falling edge of timer A1, A2, or A4 one-shot pulse
INV13 bit	Effective when INV11 is set to 1 and INV06 is set to 0	No effect

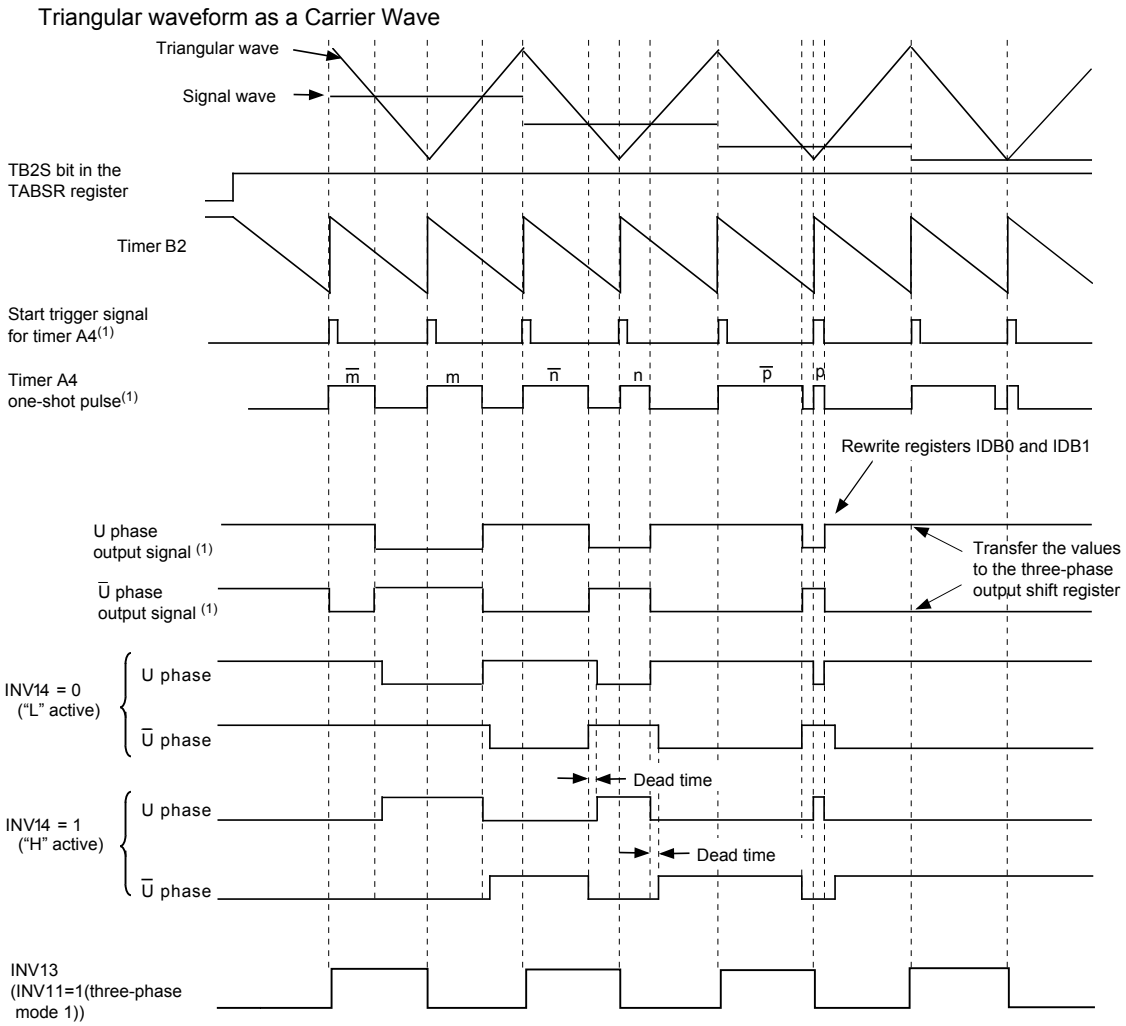
Transfer trigger: Timer B2 underflow, write to the INV07 bit or write to the TB2 register when the INV10 bit is set to 1.

9: If the INV06 bit is set to 1, set the INV11 bit to 0 (three-phase mode 0) and set the PWCON bit to 0 (timer B2 reloaded by a timer B2 underflow)

10. When the PFCi (i = 0 to 5) bit in the PFCR register is set to 1 (three-phase PWM output), individual pins are enabled to output.

Figure 12.26 INVC0 Register

The three-phase motor control timer function is enabled by setting the INV02 bit in the INVC0 register to 1. When this function is on, timer B2 is used to control the carrier wave, and timers A4, A1 and A2 are used to control three-phase PWM outputs (U, $\overline{U}$, V, $\overline{V}$, W and $\overline{W}$). The dead time is controlled by a dedicated dead-time timer. **Figure 12.33** shows the example of triangular modulation waveform, and **Figure 12.34** shows the example of sawtooth modulation waveform.



NOTE:
1. Internal signals. See **Figure 12.25**.

The above applies under the following conditions:
INVC0 = 00XX11XX2 (X varies depending on each system) and INVC1 = 010XXXX02.
Examples of PWM output change are:

(1)When INV11 = 1 (three-phase mode 1)
- INV01 = 0 and ICTB2 = 2$_{16}$ (the timer B2 interrupt is generated every two times the timer B2 underflows), or INV01 = 1, INV00 = 1, and ICTB2=1$_{16}$ (the timer B2 interrupt is generated at the falling edge of the timer A1 reload control signal.)
- Default value of the timer: TA41 = $\overline{m}$, TA4 = m. Registers TA4 and TA41 are changed whenever the timer B2 interrupt is generated. First time, TA41 = $\overline{n}$, TA4 = n. Second time, TA41 = $\overline{p}$, TA4 = p.
- Default values of registers IDB0 and IDB1: DU0 = 1, DUB0 = 0, DU1 = 0, DUB1 = 1. They are changed to DU0 = 1, DUB0 = 0, DU1= 1 and DUB1 = 0 when the third timer B2 interrupt is generated.

(2)When INV11 = 0 (three-phase mode 0)
- INV01 = 0, ICTB2 = 1$_{16}$ (the timer B2 interrupt is generated whenever timer B2 underflows)
- Default value of the timer: TA4 = $\overline{m}$. The TA4 register is changed whenever the timer B2 interrupt is generated. First time: TA4 = m. Second tim:, TA4 = $\overline{n}$. Third time: TA4 = n. Fourth time: TA4 = $\overline{p}$. Fifth time: TA4 = p.
- Default values of registers IDB0 and IDB1: DU0 = 1, DUB0 = 0, DU1 = 0, DUB1 = 1. They are changed to DU0 = 1, DUB0 = 0, DU1 = 1, and DUB1 = 0 when the sixth timer B2 interrupt is generated.

The value written to registers TA4 and TA41 becomes effective at the rising edge of the timer A1 reload control signal.

Figure 12.33 Triangular Wave Modulation Operation

12.3.1 Position-Data-Retain Function

This function is used to retain the position data synchronously with the three-phase waveform output.There are three position-data input pins for U, V, and W phases.

A trigger to retain the position data (hereafter, this trigger is referred to as "retain trigger") can be selected by the PDRT bit in the PDRF register. This bit selects the retain trigger to be the falling edge of each positive phase, or the rising edge of each positive phase.

12.3.1.1 Operation of the Position-data-retain Function

Figure 12.35 shows a usage example of the position-data-retain function (U phase) when the retain trigger is selected as the falling edge of the positive signal.

(1) At the falling edge of the U-phase waveform ouput, the state at pin IDU is transferred to the PDRU bit in the PDRF register.

(2) Until the next falling edge of the Uphase waveform output,the above value is retained.

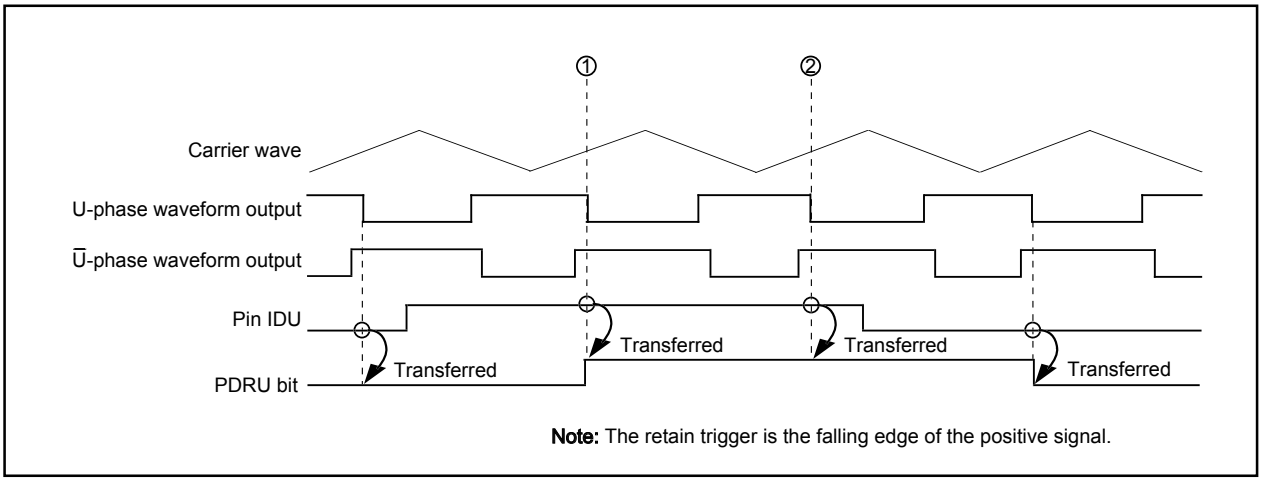
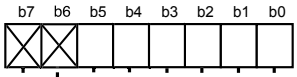


Figure 12.35 Usage Example of Position-data-retain Function (U phase)

Port Function Control Register (1)

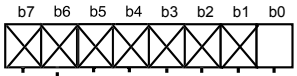


Symbol	Address	After Reset
PFCR	0358_{16}	$0011\ 1111_2$

Bit Symbol	Bit Name	Function	RW
PFC0	Port $P8_0$ output function select bit	0: Input/Output port $P8_0$ 1: Three-phase PWM output (U phase output)	RW
PFC1	Port $P8_1$ output function select bit	0: Input/Output port $P8_1$ 1: Three-phase PWM output ($\overline{U}$ phase output)	RW
PFC2	Port $P7_2$ output function select bit	0: Input/Output port $P7_2$ 1: Three-phase PWM output (V phase output)	RW
PFC3	Port $P7_3$ output function select bit	0: Input/Output port $P7_3$ 1: Three-phase PWM output ($\overline{V}$ phase output)	RW
PFC4	Port $P7_4$ output function select bit	0: Input/Output port $P7_4$ 1: Three-phase PWM output (W phase output)	RW
PFC5	Port $P7_5$ output function select bit	0: Input/Output port $P7_5$ 1: Three-phase PWM output ($\overline{W}$ phase output)	RW
$\overline{(b7\text{-}b6)}$	Nothing is assigned. When write, set to 0. When read, these contents are 0		—

NOTE:
1. This register is valid only when the INVC03 bit in the INVC0 register is 1 (Three-phase motor control timer.

Three-phase Protect Control Register



Symbol	Address	After Reset
TPRC	$025A_{16}$	00_{16}

Bit Symbol	Bit Name	Function	RW
TPRC0	Three-phase protect control bit	Enable write to PFCR register 0: Write protected 1: Write enabled	RW
$\overline{(b7\text{-}b1)}$	Nothing is assigned. If necessary, set to 0. When read, the contents are 0		—

Figure 12.38 PFCR Register, and TPRC Register

Interrupt Request Register (1)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	After Reset
G1IR	0330_{16}	Undefined

Bit Symbol	Bit Name	Function	RW
G1IR0	Interrupt request, Ch0	0 : No interrupt request 1 : Interrupt requested	RW
G1IR1	Interrupt request, Ch1		RW
G1IR2	Interrupt request, Ch2		RW
G1IR3	Interrupt request, Ch3		RW
G1IR4	Interrupt request, Ch4		RW
G1IR5	Interrupt request, Ch5		RW
G1IR6	Interrupt request, Ch6		RW
G1IR7	Interrupt request, Ch7		RW

NOTE:
1. When writing 0 to each bit in the G1IR register, use the following instruction:
AND, BCLR

Figure 13.9 G1IR Register

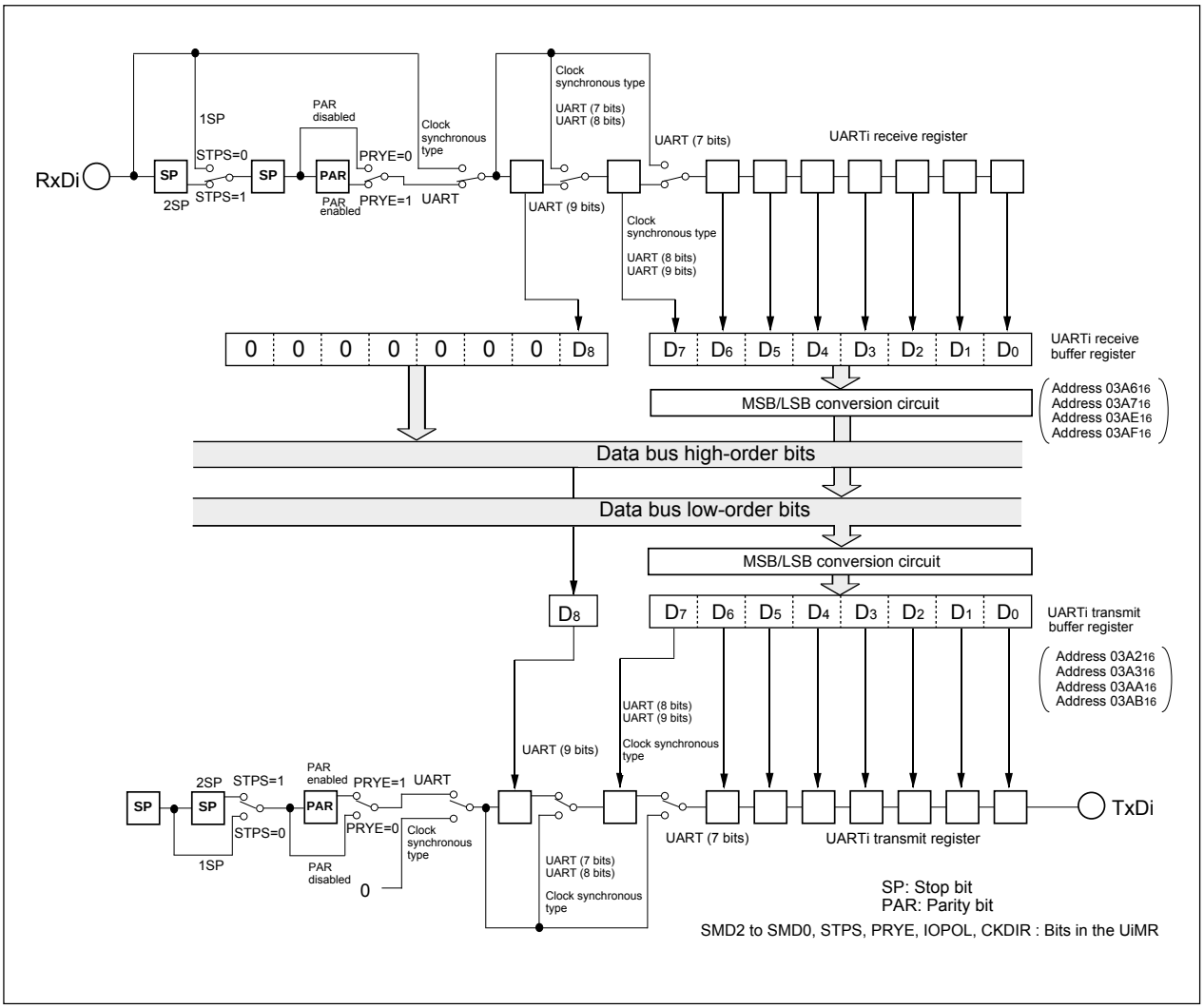


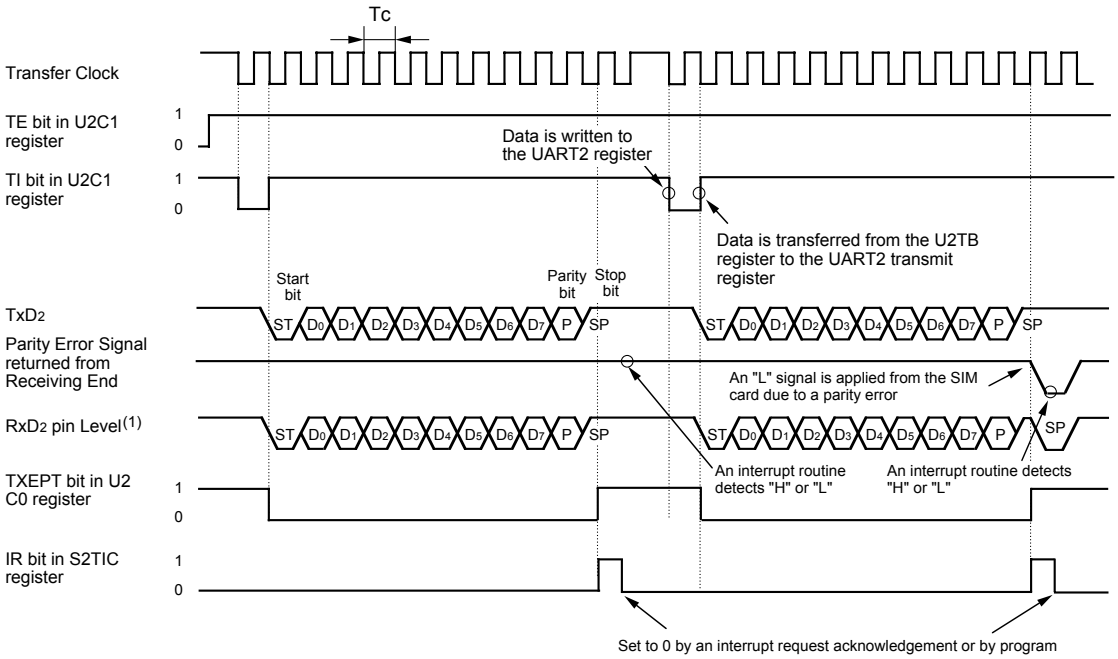
Figure 14.2 Block diagram of UARTi (i = 0, 1) transmit/receive unit

Table 14.13 I^2C bus Mode Functions

Function	Clock synchronous serial I/O mode (SMD2 to SMD0 = 001₂, IICM = 0)	I^2C bus mode (SMD2 to SMD0 = 010₂, IICM = 1)			
		IICM2 = 0 (NACK/ACK interrupt)		IICM2 = 1 (UART transmit/ receive interrupt)	
		CKPH = 0 (no clock delay)	CKPH = 1 (clock delay)	CKPH = 0 (no clock delay)	CKPH = 1 (clock delay)
Interrupt source for number 10 (1) (See **Fig.14.23**)	–	Start condition detection or stop condition detection (Refer to **Table 14.14**)			
Interrupt source for number 15 (1) (See **Fig.14.23**)	UART2 transmit operation - transmit operation is started or completed (selected by U2IRS	No acknowledgment detection (NACK) - at the rising edge of 9th bit of SCL2		UART2 transmit operation - at the rising edge of 9th bit of SCL2	UART2 transmit operation - at the next falling edge after the 9th bit of SCL2
Interrupt source for number 16 (1) (See **Fig.14.23**)	UART2 receive timing - when 8th bit is received, CKPOL = 0 (rising edge) CKPOL = 1 (falling edge)	Acknowledgment detection (ACK) - at the rising edge of 9th bit of SCL2		UART2 receive operation - at the falling edge of 9th bit of SCL2	
Data transfer timing from the UART receive shift register to the U2RB register	CKPOL = 0 (rising edge) CKPOL = 1 (falling edge)	At the rising edge of 9th bit of SCL2		Falling edge of 9th bit of SCL2	Falling edge and rising edge of 9th bit of SCL2
UART2 transmit output delay	No delay	Delay			
Function of P70	TxD2 output	SDA2 input and output			
Function of P71	RxD2 input	SCL2 input and output			
Function of P72	Select CLK2 input or output	- (Not used in I^2C bus mode)			
Noise filter width	15ns	200n			
Reading RxD2, SCL2 pin levels	Can be read if the corresponding port direction bit is set to 0	Can be read regardless of the corresponding port direction bit			
Default value of TxD2, SDA2 output	CKPOL = 0 (H) CKPOL = 1 (L)	Value set in the port register before entering I^2C bus mode(1)			
SCL2 default and end values	–	Acknowledgment detection (ACK)		UART2 receive operation - at the falling edge of 9th bit of SCL2	
DMA1 source (See **Fig.14.23**)	UART2 receive operation	H	L	H	L
Storing receive data	1st to 8th bits are stored into bits 7 to 0 in the U2RB register	1st to 8th bits are stored into bits 7 to 0 in the U2RB register		1st to 7th bits are stored into the bit 6 to bit 0 in the U2RB register, with 8th bit stored in the bit 8 in the U2RB register	1st to 8th bits are stored into bits 7 to 0 in the U2RB register(3)
Reading receive data	The U2RB register status is read				Bit 6 to bit 0 in the U2RB register are read as bit 7 to bit 1. Bit 8 in the U2RB register is read as bit 0 (4)

NOTES:
1. If the interrupt source is changed, the IR bit in the interrupt control register for the changed interrupt may be set to 1 (interrupt requested). (Refer to “Interrupts” in Precautions.) If any of the following bits are changed, the interrupt source, the interrupt timing, etc. will be changed also. Therefore, set the IR bit to 0 (interrupt not requested) after those bits are changed.
   Bits SMD2 to SMD0 in the U2MR register, the IICM bit in the U2SMR register, the IICM2 bit in the U2SMR2 register, the CKPH bit in the U2SMR3 register
2. Set the default value of the SDA2 output when bits SMD2 to SMD0 in the U2MR register are set to 000₂ (serial I/O disabled).
3. Second data transfer to the U2RB register (at the rising edge of the ninth bit of SCL2)
4. First data transfer to the U2RB register (at falling edge of the ninth bit of SCL2)

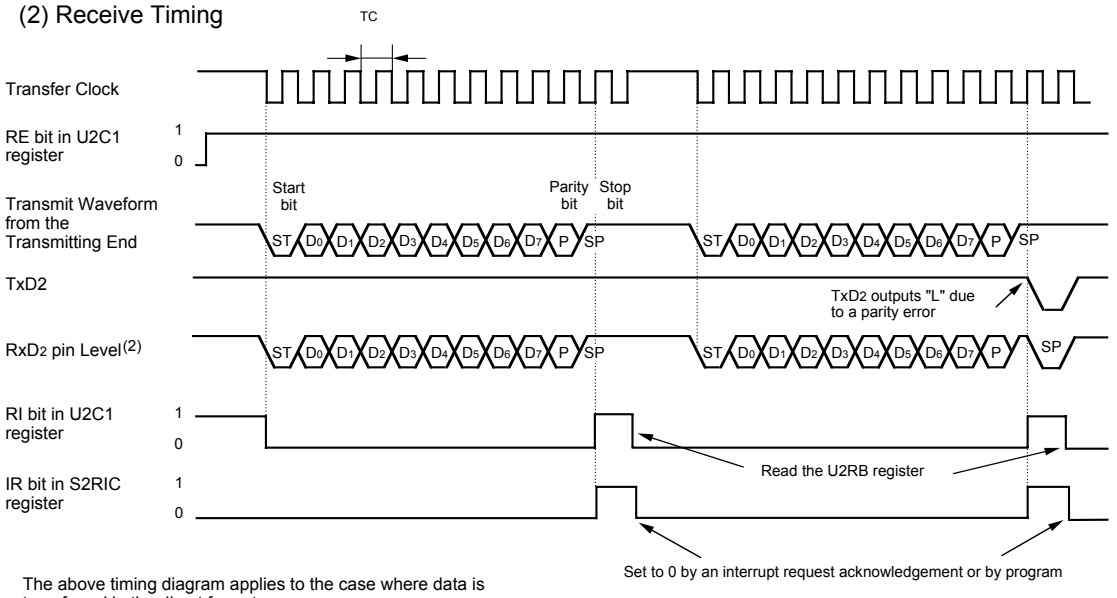
(1) Transmit Timing



The above timing diagram applies to the case where data is transferred in the direct format.
- U2MR register STPS bit = 0 (1 stop bit)
- U2MR register PRY bit = 1 (even)
- U2C0 register UFORM bit = 0 (LSB first)
- U2C1 register U2LCH bit = 0 (no reverse)
- U2C1 register U2IRSCH bit = 1 (transmit is completed)

Tc = 16 (n + 1) / fi or 16 (n + 1) / fEXT
fi : frequency of U2BRG count source (f1SIO, f2SIO, f8SIO, f32SIO)
fEXT : frequency of U2BRG count source (external clock)
n : value set to U2BRG

(2) Receive Timing



The above timing diagram applies to the case where data is transferred in the direct format.
- U2MR register STPS bit = 0 (1 stop bit)
- U2MR register PRY bit = 1 (even)
- U2C0 register UFORM bit = 0 (LSB first)
- U2C1 register U2LCH bit = 0 (no reverse)
- U2C1 register U2IRSCH bit = 1 (transmit is completed)

Tc = 16 (n + 1) / fi or 16 (n + 1) / fEXT
fi : frequency of U2BRG count source (f1SIO, f2SIO, f8SIO, f32SIO)
fEXT : frequency of U2BRG count source (external clock)
n : value set to U2BRG

NOTES:
1. Because TxD2 and RxD2 are connected, this is composite waveform consisting of the TxD2 output and the parity error signal sent back from receiver.
2. Because TxD2 and RxD2 are connected, this is composite waveform consisting of the transmitter's transmit waveform and the parity error signal received.

Figure 14.31 Transmit and Receive Timing in SIM Mode

A/D Control Register 0 (1)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset
			0	0				ADCON0	03D6$_{16}$	00000XXX$_2$

Bit Symbol	Bit Name	Function	RW
CH0	Analog input pin select bit (2, 3)	b2 b1 b0 0 0 0: Select AN0 0 0 1: Select AN1 0 1 0: Select AN2 0 1 1: Select AN3 1 0 0: Select AN4 1 0 1: Select AN5 1 1 0: Select AN6 1 1 1: Select AN7	RW
CH1			RW
CH2			RW
MD0	A/D operation mode select bit 0 (3)	b4 b3 0 0: One-shot mode or delayed trigger mode 0,1	RW
MD1			RW
TRG	Trigger select bit	0: Software trigger 1: Hardware trigger ($\overline{\text{ADTRG}}$ trigger)	RW
ADST	A/D conversion start flag	0: A/D conversion disabled 1: A/D conversion started	RW
CKS0	Frequency select bit 0	See **Table 15.2**	RW

NOTES:
1. If the ADCON0 register is rewritten during A/D conversion, the conversion result will be undefined.
2. AN00 to AN07, AN20 to AN27, and AN30 to AN32 can be used in the same way as AN0 to AN7. Use bits ADGSEL1 and ADGSEL 0 in the ADCON2 register to select the desired pin.
3. After rewriting bits MD1 and MD0, set bits CH2 to CH0 over again using an another instruction.

A/D Control Register 1 (1)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset
X	X	1			0			ADCON1	03D7$_{16}$	00$_{16}$

Bit Symbol	Bit Name	Function	RW
SCAN0	A/D sweep pin select bit	Invalid in one-shot mode	RW
SCAN1			RW
MD2	A/D operation mode select bit 1	0: Other than repeat sweep mode 1	RW
BITS	8/10-bit mode select bit	0: 8-bit mode 1: 10-bit mode	RW
CKS1	Frequency select bit 1	See **Table 15.2**	RW
VCUT	Vref connect Bit (2)	1: Vref connected	RW
$\overline{\text{(b7-b6)}}$	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

NOTES:
1. If the ADCON1 register is rewritten during A/D conversion, the conversion result will be undefined.
2. If the VCUT bit is reset from 0 (Vref unconnected) to 1 (Vref connected), wait for 1 µs or more before starting A/D conversion.

A/D Control Register 2(1)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset
X	X	0		0				ADCON2	03D4$_{16}$	00$_{16}$

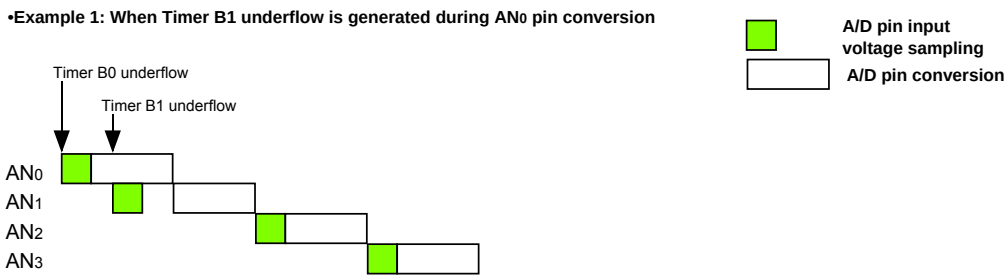
Bit Symbol	Bit Name	Function	RW
SMP	A/D conversion method select bit	0: Without sample and hold 1: With sample and hold	RW
ADGSEL0	A/D input group select bit	b2 b1 0 0: Select port P10 group 0 1: Select port P9 group 1 0: Select port P0 group 1 1: Select port P1/P9 group	RW
ADGSEL1			RW
$\overline{\text{(b3)}}$	Reserved bit	Set to 0	RW
CKS2	Frequency select bit 2	See **Table 15.2**	RW
TRG1	Trigger select bit	Set to 0 in one-shot mode	RW
$\overline{\text{(b7-b6)}}$	Nothing is assigned. If necessary, set to 0. When read, the content is 0		—

NOTE:
1. If the ADCON2 register is rewritten during A/D conversion, the conversion result will be undefined.

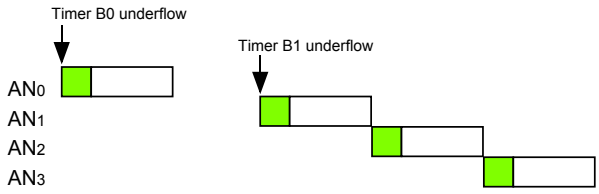
Figure 15.7 ADCON0 to ADCON2 Registers in One-Shot Mode

•Example when selecting AN_0 to AN_3 to A/D sweep pins (SCAN1 to SCAN0 = 01_2)

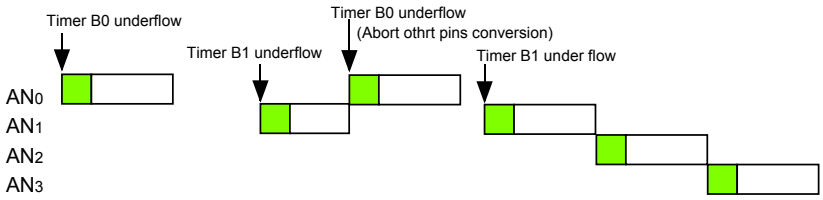
•Example 1: When Timer B1 underflow is generated during AN_0 pin conversion



•Example 2: When Timer B1 underflow is generated after AN_0 pin conversion



•Example 3: When Timer B0 underflow is generated during A/D conversion of any pins except AN_0 pin



•Example 4: When Timer B0 underflow is generated again before Timer B1 underflow is generated after Timer B0 underflow generation

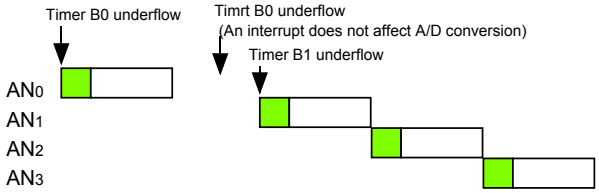


Figure 15.19 Operation Example in Delayed Trigger Mode 0

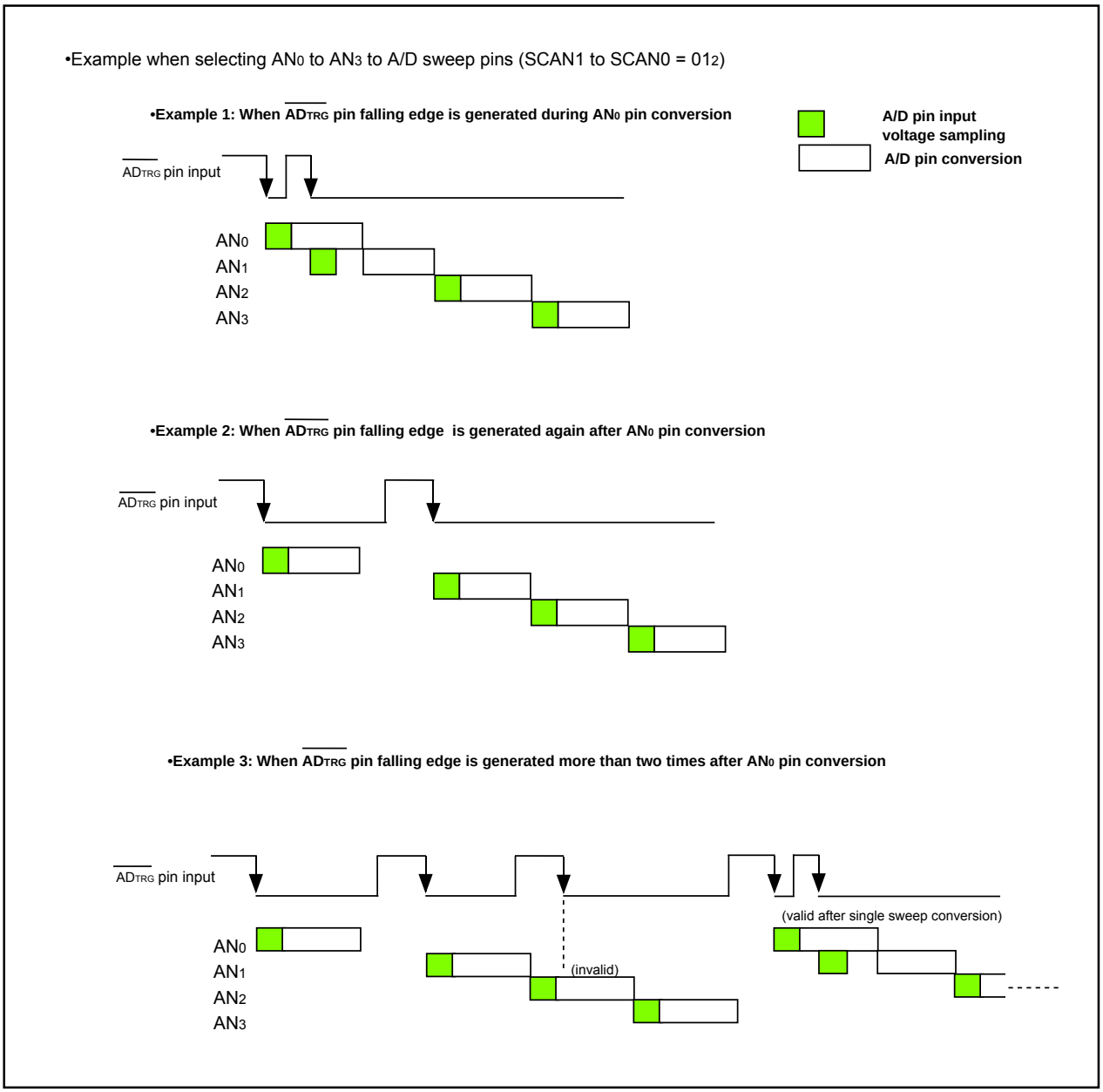


Figure 15.24 Operation Example in Delayed Trigger Mode1

21.6 Timer

21.6.1 Timer A

21.6.1.1 Timer A (Timer Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAiMR (i = 0 to 4) register and the TAi register before setting the TAiS bit in the TABSR register to 1 (count starts).
   Always make sure the TAiMR register is modified while the TAiS bit remains 0 (count stops) regardless whether after reset or not.

2. While counting is in progress, the counter value can be read out at any time by reading the TAi register. However, if the TAi register is read at the same time the counter is reloaded, the read value is always FFFF16. If the TAi register is read after setting a value in it, but before the counter starts counting, the read value is the one that has been set in the register.

3. If a low-level signal is applied to the $\overline{SD}$ pin when the IVPCR1 bit in the TB2SC register is set to 1 (three-phase output forcible cutoff by input on $\overline{SD}$ pin enabled), the TA1OUT, TA2OUT and TA4OUT pins go to a high-impedance state.

21.6.1.2 Timer A (Event Counter Mode)

1. The timer remains idle after reset. Set the mode, count source, counter value, etc. using the TAiMR (i = 0 to 4) register, the TAi register, the UDF register, bits TAZIE, TA0TGL, and TA0TGH in the ONSF register and the TRGSR register before setting the TAiS bit in the TABSR register to 1 (count starts).
   Always make sure bits TAZIE, TA0TGL, and TA0TGH in the TAiMR register, the UDF register, the ONSF register, and the TRGSR register are modified while the TAiS bit remains 0 (count stops) regardless whether after reset or not.

2. While counting is in progress, the counter value can be read out at any time by reading the TAi register. However, if the TAi register is read at the same time the counter is reloaded, the read value is always FFFF16 when the timer counter underflows and 000016 when the timer counter overflows. If the TAi register is read after setting a value in it, but before the counter starts counting, the read value is the one that has been set in the register.

3. If a low-level signal is applied to the $\overline{SD}$ pin when the IVPCR1 bit in the TB2SC register is set to 1 (three-phase output forcible cutoff by input on $\overline{SD}$ pin enabled), the TA1OUT, TA2OUT and TA4OUT pins go to a high-impedance state.

8. If the CPU reads the ADi register (i = 0 to 7) at the same time the conversion result is stored in the ADi register after completion of A/D conversion, an incorrect value may be stored in the ADi register. This problem occurs when a divide-by-n clock derived from the main clock or a subclock is selected for CPU clock.
   - When operating in one-shot, single-sweep mode, simultaneous sample sweep mode, delayed trigger mode 0 or delayed trigger mode 1:
     Check to see that A/D conversion is completed before reading the target ADi register. (Check the IR bit in the ADIC register to see if A/D conversion is completed.)
   - When operating in repeat mode or repeat sweep mode 0 or 1:
     Use the main clock for CPU clock directly without dividing it.

9. If A/D conversion is forcibly terminated while in progress by setting the ADST bit in the ADCON0 register to 0 (A/D conversion halted), the conversion result of the A/D converter is undefined. The contents of ADi registers irrelevant to A/D conversion may also become undefined. If while A/D conversion is underway the ADST bit is cleared to 0 in a program, ignore the values of all ADi registers.

10. When setting the ADST bit in the ADCON register to 0 and terminating forcefully by a program in single sweep conversion mode, A/D delayed trigger mode 0 and A/D delayed trigger mode 1 during A/D converting operation, the A/D interrupt request may be generated. If this causes a problem, set the ADST bit to 0 after an interrupt is disabled.